PENNANT: A Research Tool for Unstructured Mesh Physics on Advanced Architectures

Charles R. Ferenbaugh,
HPC-1

Novel computer architectures such as graphics processing units (GPU), many-core chips, and IBM BlueGene are becoming common in the high-performance computing (HPC) world. Existing physics application codes require major modifications to perform well on these architectures. LANL has developed the PENNANT mini-app as a research tool for finding efficient implementations of unstructured mesh physics algorithms. It contains mesh data structures and a few physics algorithms adapted from the LANL shock physics code FLAG and will provide insights on how to optimize FLAG and other similar codes for future architectures.

The HPC world is entering a major transition. Novel architectures such as GPUs, many-core chips, and IBM's BlueGene are becoming common in supercomputer clusters. Architectures such as these provide high computational performance combined with low power usage, and are likely to be used in future systems such as the "exascale" systems being discussed in the international HPC community.

These systems will pose significant challenges to all scientific software developers—introducing new programming models to manage the increased hardware complexity and requiring major rewrites of existing software. They will be particularly challenging for developers of algorithms for general unstructured meshes—that is, computational meshes containing arbitrary polygons (in 2D) or polyhedra (in 3D). An example is shown in Fig. 1. These meshes have irregular connectivity and memory usage patterns, making them more difficult to work with than meshes with more regular structure. As a result, unstructured mesh methods tend to lag behind other types of physics methods in advanced architecture research.

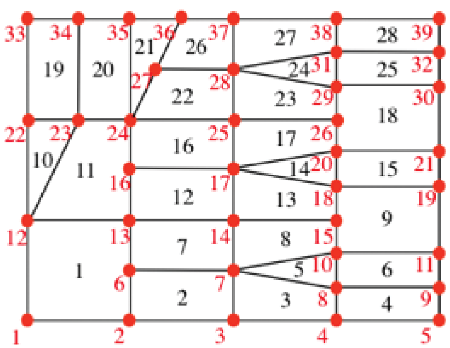

Fig. 1. Example of an unstructured mesh.

However, since unstructured mesh codes are common at LANL and elsewhere, it is important to find ways to run such codes efficiently on these architectures. As a tool for research in this area, LANL has developed a small application, or mini-app, called PENNANT [5].

PENNANT contains approximately 2200 lines of C++ source code. It implements a small subset of the physics of the LANL shock physics application FLAG. Like FLAG, it operates on general unstructured meshes (meshes containing arbitrary polygons). It currently has implementations for serial and multi-core processors–the multi-core version uses the standard OpenMP programming model. A GPU implementation using the CUDA programming language is in progress, building on previous research [4].

PENNANT provides the following basic physics capabilities from FLAG:

- Lagrangian staggered grid hydrodynamics (SGH) [2] for basic fluid flow
- Single material, gamma-law gas equation of state
- Temporary Triangular Subzoning (TTS) [3] for subzonal pressures
- Campbell-Shashkov tensor artificial viscosity [1]

These capabilities are sufficient to run a few simple test problems in shock physics.

The following PENNANT timing results were obtained on the Darwin research cluster at LANL. The nodes of this cluster have 4 12-core AMD Opteron 6168 CPUs, for a total of 48 cores, each running at 1.90 GHz. The nodes used in this study also have NVidia M2090 GPUs attached.

Two different versions of the Noh test problem [6], nohsquare and nohpoly, were used. A sample output from nohpoly is shown in Fig. 2.

Figure 3 shows timings from a scaling study done using the OpenMP version with varying numbers of threads. The dashed line shows ideal scaling of the serial version, for comparison. For both problems, the OpenMP implementation of PENNANT scales well on up to 32 cores, but starts to level off when using the full 48 cores of a Darwin node. The 32-core runs showed a speedup of 22× over the serial version.

A GPU implementation is in progress. Currently the force calculation is running on the GPU, and timings on this portion of the code show about a 14× to 17× speedup (problem-dependent) over the corresponding serial code. Past experience suggests that the full GPU version, when complete, will show a similar speedup.

PENNANT demonstrates that unstructured mesh physics can be implemented efficiently on multicore processors and GPUs. It also shows

For more information contact Charles R. Ferenbaugh at cferenba@lanl.gov.

Funding Acknowledgments
DOE, NNSA, Advanced Simulation and Computing Program; Office of Science, Advanced Scientific Computing Research Program

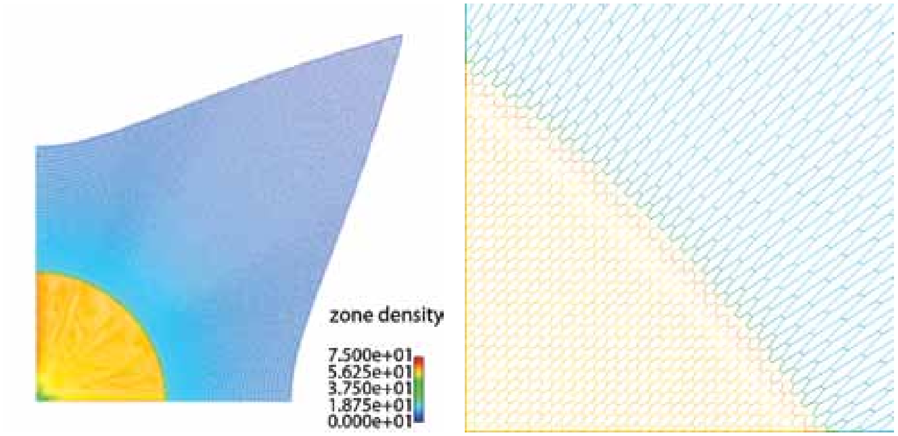

Fig. 2. PENNANT output for Noh problem on a mesh with hexagonal cells.

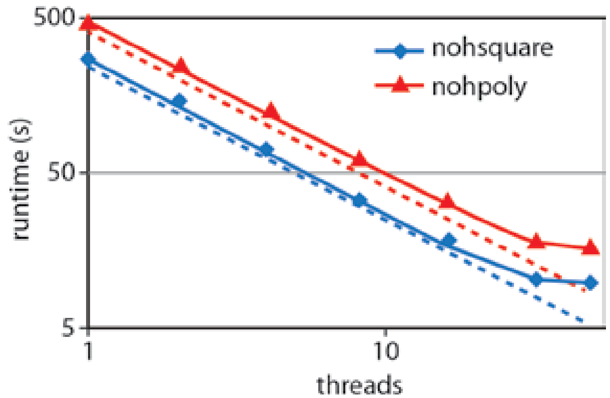

Fig. 3. PENNANT timings running for nohsquare and nohpoly test problems, using up to 48 cores.

that a small, self-contained code can capture the basic physics algorithms and data structures of larger multi-physics codes that use unstructured meshes.

Future development plans for PENNANT at LANL include

- additional test problems;
- additional optimizations for serial and multicore versions;
- multi-node version using the Message-Passing Interface (MPI) library to distribute work across the nodes in a cluster;
- GPU versions using other programming models (OpenCL and/or OpenACC);
- and testing on Intel Many-Integrated-Cores (MIC) architecture.

PENNANT will also be made available to other research sites, hardware vendors, and compiler vendors as a research tool to help make unstructured mesh algorithms run more effectively on future hardware platforms and programming models.

[1] Campbell, J. and M. Shashkov, *J Comput Phys* **172**, 739 (2001).

[2] Caramana, E.J. et al., *J Comput Phys* **146**, 227 (1998).

[3] Caramana, E.J. and M.J. Shashkov, *J Comput Phys* **142,** 521 (1998).

[4] Ferenbaugh, C.R, “A Copomparison of GPU Strategies for Unstructured Mesh Physics,” *Concurrency and Computation: Practice and Experience,* in press (2012).

[5] Ferenbaugh, C.R., “The PENNANT Mini-App: Unstructured Mesh Hydrodynamics for Advanced Architectures,” *Proceedings of the Nuclear Explosives Code Developers Conference* (2012).

[6] Noh, W.F., *J Comput Phys* **72**, 78 (1987).